howstuffworks
it's good to know.

EXPLANATIONS
- **Auto**
- **Computer**
  - **Hardware**
  - **Internet**
  - **Peripherals**
  - **Security**
  - **Software**
- **Electronics**
- **Entertainment**
- **Health**
- **Home**
- **Money**
- **People**
- **Science**
- **Travel**

EXPERT REVIEWS

Consumer Guide Auto

Consumer Guide Products

Mobil Travel Guide

OPINIONS

Member Home

Log In/Register

PRICES

Shop HowStuffWorks

VIDEO CENTER

Search Video Center

RELATED AD CATEGORIES

Internet DNS Server

Network Server

Server IP Address

Web Server

Windows Server

Main > Computer > **Internet**
PRINT   EMAIL

How Web Servers Work

by Marshall Brain

**Inside This Article**

Protocols

Once a client has connected to a service on a particular port, it accesses the service using a specific protocol. The **protocol** is the pre-defined way that someone who wants to use a service talks with that service. The "someone" could be a person, but more often it is a computer program like a Web browser. Protocols are often text, and simply describe how the client and server will have their conversation.

Perhaps the simplest protocol is the **daytime protocol**. If you connect to port 13 on a machine that supports a daytime server, the server will send you its impression of the current date and time and then close the connection. The protocol is, "If you connect to me, I will send you the date and time and then disconnect." Most UNIX machines support this server. If you would like to try it out, you can connect to one with the Telnet application. In UNIX, the session would look like this:

```
%telnet web67.ntx.net 13
Trying 216.27.61.137...
Connected to web67.ntx.net.
Escape character is '^]'.
Sun Oct 25 08:34:06 1998
Connection closed by foreign host.
```

ATTACHMENT ᴀ
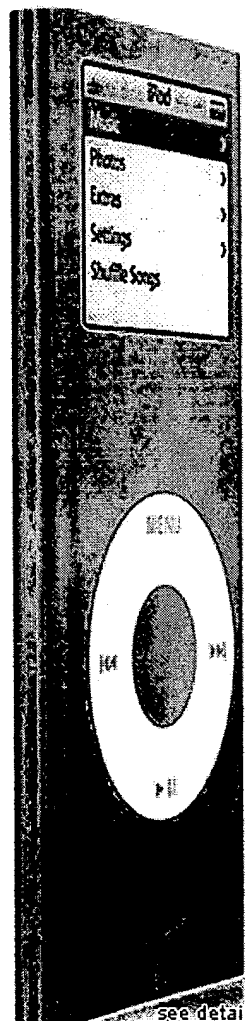
On a Windows machine, you can access this server by typing "telnet web67.ntx.net 13" at the MSDOS prompt.

In this example, web67.ntx.net is the server's UNIX machine, and 13 is the port number for the daytime service. The Telnet application connects to port 13 (telnet naturally connects to port 23, but you can direct it to connect to any port), then the server sends the date and time and disconnects. Most versions of Telnet allow you to specify a port number, so you can try this using whatever version of Telnet you have available on your machine.

Most protocols are more involved than daytime and are specified in Request for Comment (RFC) documents that are publicly available (see http://sunsite.auc.dk/RFC/ for a nice archive of all RFCs). Every Web server on the Internet conforms to the HTTP protocol, summarized nicely in The Original HTTP as defined in 1991. The most basic form of the protocol understood by an HTTP server involves just one command: GET. If you connect to a server that understands the HTTP protocol and tell it to "GET filename," the server will respond by sending you the contents of the named file and then disconnecting. Here's a typical session:

```
%telnet www.howstuffworks.com 80
Trying 216.27.61.137...
Connected to howstuffworks.com.
Escape character is '^]'.
GET http://www.howstuffworks.com/
<html>
<head>
<title>Welcome to How Stuff Works</title>
  . . .
</body>
</html>
Connection closed by foreign host.
```

In the original HTTP protocol, all you would have sent was the actual filename, such as "/" or "/web-server.htm." The protocol was later modified to handle the sending of the complete URL. This has allowed companies that host **virtual domains**, where many domains live on a single machine, to use one IP address for all of the domains they host. It turns out that hundreds of domains are hosted on 209.116.69.66 -- the HowStuffWorks IP address.

< PREVIOUS | INTRO | NEXT >

Inside This Article

PAGE TOOLS

howstuffworks
it's good to know.

Main > Computer > **Internet**
PRINT   EMAIL

How Web Pages Work

by Marshall Brain

Inside This Article

1. Introduction to How Web Pages Work
2. **Setting the Stage**
3. Basic HTML Formatting Tags
4. Creating Tables
5. Creating Frames
6. Images
7. Getting Your Page Online
8. Promoting Your Site
9. Ideas
10.
Lots More Information
11.
See all **Internet** articles

Setting the Stage

At this moment, it is nearly guaranteed that:

- You are sitting at your computer.
- You are using a **Web browser** to read this page, and that browser could be Microsoft Internet Explorer, Firefox or maybe Netscape.
- You want to learn how Web pages work, and perhaps learn the art of creating your own pages.

Because you are sitting at a computer, you have a Web browser and you possess the desire to learn, you have everything you need to get started. You can learn HTML and experiment with your Web browser to find out how to create any kind of Web page you can imagine.

In order to talk about Web pages and how they work, you will want to understand four simple terms (and if some of this sounds like technical mumbo-jumbo the first time you read it, don't worry):

- **Web page** - A Web page is a simple text file that contains not only text, but also a set of **HTML tags** that describe how the text should be formatted when a browser displays it on the screen. The tags are simple instructions that tell the Web browser how the page should look when it is displayed. The tags tell the browser to do things like change the font size or color, or arrange things in columns. The Web browser **interprets** these tags to decide how to format the text onto the screen.

- **HTML** - HTML stands for **Hyper Text Markup Language**. A "markup language" is a computer language that describes how a page should be formatted. If all you want to do is display a long string of black

and white text with no formatting, then you don't need HTML. But if you want to change fonts, add colors, create headlines and embed graphics in your page, HTML is the language you use to do it.

- **Web browser** - A Web browser, like Netscape Navigator or Microsoft Internet Explorer, is a **computer program** (also known as a **software application**, or simply an **application**) that does two things:

  - A Web browser knows how to go to a **Web server** on the Internet and request a page, so that the browser can pull the page through the network and into your machine.
  - A Web browser knows how to **interpret** the set of **HTML tags** within the page in order to display the page on your screen as the page's creator intended it to be viewed.

Your browser connects to a server and requests a page.

The server sends back the requested page.

Your machine running a Web browser

Server machine running a Web server
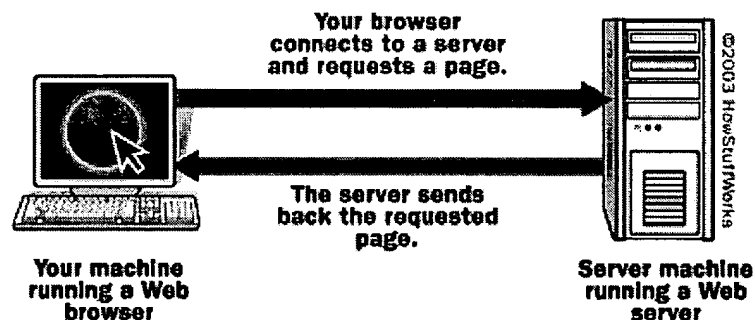
- **Web server** - A Web server is a piece of computer software that can respond to a browser's request for a page, and deliver the page to the Web browser through the Internet. You can think of a Web server as an apartment complex, with each apartment housing someone's Web page. In order to store your page in the complex, you need to pay rent on the space. Pages that live in this complex can be displayed to and viewed by anyone all over the world. Your landlord is called your **host**, and your rent is usually called your **hosting charge**. Every day, there are millions of Web servers delivering pages to the browsers of tens of millions of people through the network we call the Internet. Read How Web Servers Work for details on this process.

It is extremely easy to experiment with Web pages without using a server. Your browser can view the Web pages you create from your personal machine. Once you understand how to create your own pages, it is likely that you will want to put them "out on a server," so that people around the world can load your pages and read them. We will talk about how to do that at the end of this article.

Viewing the Source

Let's take a look at the "guts" of a Web page. This is the original text and **HTML tags** typed by the author and interpreted by the browser to produce the Web page you actually SEE on the Internet. With your mouse, right-click on any blank portion of this page and choose "View Source." A new window will appear, displaying words and characters, some of which may look pretty technical and foreign. These words and characters are, collectively, the HTML programming **code** you are about to learn. Each element within that code is known as an HTML tag. Don't be intimidated by how complex it looks -- you'll be surprised at how easy it really is. When you are done looking at the page's source code, simply close out the source page to return to this article.

You can look "behind the scenes" of almost any page on the Internet this way. When you become more involved in Web designing, you'll probably find yourself viewing the sources of complicated Web pages in order to learn the codes that the authors, or **page designers**, used to create such interesting arrangements.

Now, let's learn what many of the tags mean, and start creating a simple page.

The HTML Tag

An **HTML tag** is a code element that tells the Web browser what to do with your text. Each tag will appear as letters or words between a < (less than sign) and a > (greater than sign).

Example: <center>, <body>

To tell the Web browser to "end" doing what you just told it to do, a forward slash is used in the closing tag:

Example: </center>, </body>

Most tags come in matched "beginning" and "ending" pairs, but this is not an absolute rule.

Any Web page you create will contain the following tags at the start of the page:

- <HTML>: tells the Web browser that this is the beginning of an HTML document
- <HEAD>: tells that Web browser that this is the header for the page (you'll learn later what goes between "HEAD" tags)
- <TITLE>: tells the Web browser that this is the title of the page
- <BODY>: tells the Web browser that this is the beginning of the Web page content -- everything you want to say and see on your page will follow this tag.

The tags needed to end any Web page are:

- </BODY>
- </HTML>

Creating a Simple Page

There are many ways to create Web pages. Hundreds of companies have created tools to help with the process in one way or another. Our goal here, however, is to understand how Web pages really work, rather than have a tool hide the process from us. Therefore, we will use the simplest tools possible -- the ones already on your machine.

On your machine you have a program, or **application**, that can create simple text files. On Windows machines, this application is called **Notepad**. On a Macintosh, this program is called **SimpleText**. If you cannot find these programs, it is also possible to use a basic word processing program, like WordPerfect or Microsoft Word.

Note:
- In a Windows 95/98 environment, click the Start button, click Programs, click Accessories and click Notepad.
- In a Windows 3.1 environment, click Accessories (in the Program

Manager) and click Notepad.
- In a Macintosh environment, click Macintosh HD, click Applications and click SimpleText.

Once you have the proper program open on your screen, type (or cut-and-paste) the following HTML text into the window:

```
<html>
  <head>
    <title>My First Page</title>
  </head>
  <body>
    Hello there. This is my first page!
  </body>
</html>
```

Whether you put the tags and text side-by-side, row-by-row or indented will not make a difference in how the text is displayed in a browser window. Whether you use uppercase or lowercase letters within your tags also does not make a difference.

Now you need to save this file somewhere so that you can find it in a moment. Save it to the desktop, or, better yet, to a directory that you set up to hold all of the pages you are going to create. Save it to the filename **first.html**.

Next, open the page in your Web browser (e.g., Microsoft Internet Explorer or Netscape Navigator). All Web browsers have a way to open a file stored on the local machine. In Internet Explorer and Netscape, select Open File from the File menu at the top of the window. Open the file **first.html**. When you open it in your browser, it will look something like this:
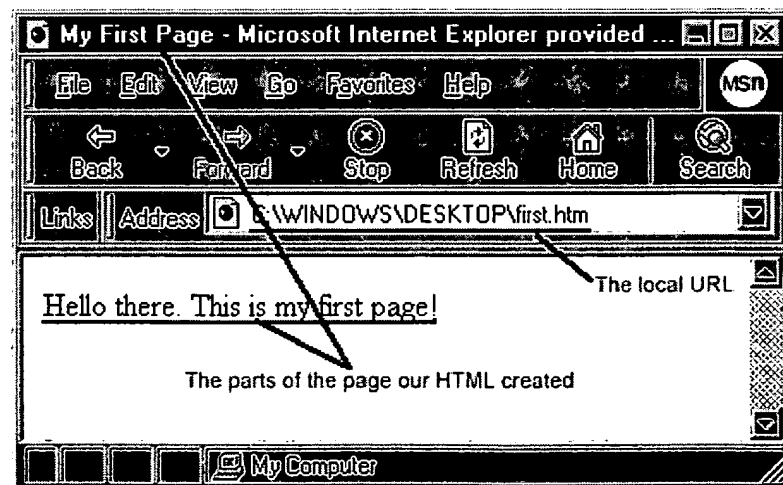
Three things are identified in this figure:

- You can see that the page has the title "My First Page."
- You can see that the page's body contains the words "Hello there. This is my first page!"
- You can see that the URL being displayed in the address window is **C:\WINDOWS\DESKTOP\first.html** from the local hard disk, rather than the usual **http://...** that a URL would contain if we were